

(12) PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 199857832 B2**
(10) Patent No. **739924**

(54) Title
Method of transferring media files over a communications network

(51) ⁶ International Patent Classification(s)
H04N 007/173

(21) Application No: 199857832 (22) Application Date: 1998.01.26

(87) WIPO No: WO98/33320

(30) Priority Data

(31) Number	(32) Date	(33) Country
314150	1997.01.29	NZ

(43) Publication Date : 1998.08.18

(43) Publication Journal Date : 1998.10.01

(44) Accepted Journal Date : 2001.10.25

(71) Applicant(s)
Tangozebra Limited

(72) Inventor(s)
Stewart Neil Everett; James Barrs; James Augustine Booth; Ian David Montgomery

(74) Agent/Attorney
ALLENS ARTHUR ROBINSON,GPO BOX 1776Q,MELBOURNE VIC 3001

(56) Related Art
EP 702493

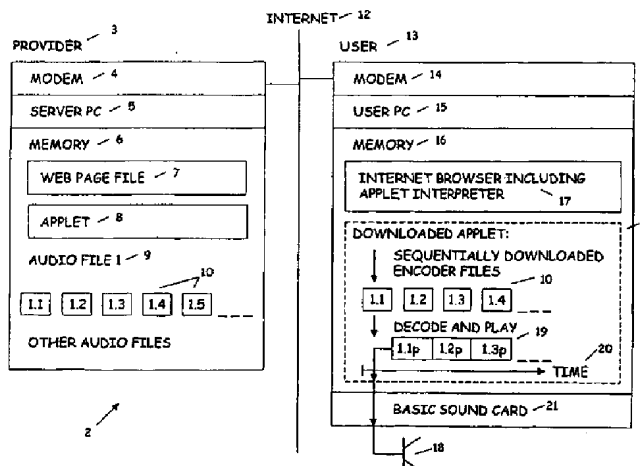
OPI DATE 18/08/98 APPLN. ID 57832/98
AOJP DATE 01/10/98 PCT NUMBER PCT/NZ98/00005



AU9857832

(51) International Patent Classification 6 : H04N 7/13		(11) International Publication Number: WO 98/33320	
A1		(43) International Publication Date: 30 July 1998 (30.07.98)	
(21) International Application Number: PCT/NZ98/00005		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 26 January 1998 (26.01.98)		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.	
(30) Priority Data: 314150 29 January 1997 (29.01.97) NZ		(71) Tangozebra Limited	
(71) Applicant (for all designated States except US): WEST-WALKER, Francis, Nigel [NZ/NZ]; McCabe and Company, Fraser House, 3rd floor, 160-162 Willis Street, Wellington (NZ).		(71) Tangozebra Limited 21 Market Place, Cirencester Gloucestershire GL7 2RP United Kingdom 3rd Flr, 14-18 Great Titchfield St. London W1P 7AB (GB)	
(71)(72) Applicants and Inventors: EVERETT, Stewart, Neil [GB/GB]; 24 Roberts Close, Stratton, Cirencester, Gloucestershire GL7 2RP (GB). BARRS, James [GB/GB]; 2 Roberts Close, Stratton, Cirencester, Gloucestershire G1 2RP (GB). BOOTH, James, Augustine [GB/GB]; Lower Manor, Frampton Mansell, Stroud, Gloucestershire GL6 8JG (GB). MONTGOMERY, Ian, David [GB/GB]; Puck Miller House, Frampton Mansell, Stroud, Gloucestershire GL6 8JQ (GB).		AUSTRALIAN PATENT OFFICE SEC 104 AUSTRALIAN PATENT OFFICE SEC 113	
(74) Agent: WEST-WALKER, Francis, Nigel; McCabe and Company, Fraser House, 3rd floor, 160-162 Willis Street, Wellington (NZ).			

(54) Title: METHOD OF TRANSFERRING MEDIA FILES OVER A COMMUNICATIONS NETWORK



(57) Abstract

A method of transferring media files over a communications network, typically the Internet. The files are divided by the provider computer into a series of encoded files which are maintained in the provider computer and are transferred over the communications network in a specific sequence to receiving computer. A user loadable program is also maintained in the provider computer and that together with all file types contributing to the content of the communication are also transferred over the communications network to the receiving computer. The received files can then be reproduced by the receiving computer in the correct sequence.

1

**METHOD OF TRANSFERRING MEDIA FILES
OVER A COMMUNICATIONS NETWORK**

FIELD OF THE INVENTION

5 The present invention relates to a method of transferring and reproducing media and other file types over a communications network in a prearranged sequence/order and the managed reproduction of said file types along a sequencing time line where reproduction is intelligently managed in accordance with the fluctuations in the throughput speed of the network and the effect this has on the availability of subsequent files. In particular but not exclusively the present invention relates to a computerised method of
10 transferring and reproduction media or other file types over the Internet where the file types are component parts of a greater arrangement and reproducing of said files happens in line with the reconstruction of the greater arrangement 'on the fly' in the Internet browser software.

BACKGROUND OF THE INVENTION

15 In this specification, where a document, act or item of knowledge is referred to or discussed, this reference or discussion is not an admission that the document, act or item of knowledge or any combination thereof was at the priority date publicly available, known to the public, part of the common general knowledge or known to be relevant to an attempt to
20 solve any problem with which this specification is concerned.

25 With regard to media files, there are currently several available computerised methods of transferring media files over communications networks, such as the Internet, but they all have specific disadvantages. Some known downloadable audio file techniques rely on sending an audio file as a package of data or digitised audio, and the receiver must wait until the media file is fully loaded before complete reproduction of the file can commence. Particularly when applied to the Internet, the length of time taken to download a media file can be such that the user might well decide to disconnect before the media file has downloaded and can be played.



clrm 10993855v1

1A

Live stream type methods of communicating media files such as Real Audio or Shockwave for example, transfer compressed audio files which are decoded and played as the receiver receives them with the aid of buffering techniques. However these methods of
5 transferring compressed audio files normally require stable areas of bandwidth and appropriate decoding software at the receiver end. Another disadvantage with Real Audio systems is that they require a minimum of a 28.8 Kbps modem for adequate sound reproduction. Such systems are generally and best used for broadcasting by radio stations for concert broadcasts

2000
10
20



clrm 10993855v1

or for video conferencing and are not suited for being readily incorporated into an Internet World Wide Web site as, with today's available bandwidth, the Internet as a whole could not support many simultaneous streams of continuous audio/video data coming off a significant number of web sites. Streaming media solutions offer certain facilities to the Internet developer but lack the ability to time in, interact with the user, track statistics and make use of available bandwidth.

Macromedia Flash is a recent and popular method for introducing entertaining multimedia to web sites and uses the idea of the time line to play a combination of image and sound files. However, Flash relies heavily on using vector graphics and very small audio files. As Flash arrangements largely need to be preinstalled into the browser prior to running they have to be kept to a minimum size and are not able to intelligently continue to build within the browser. 'Flash' web sites tend to have a similar feel due to the nature of vector images and as such are only suitable for certain types of Internet campaign and are not ideally suited for integrating into existing web sites.

SMIL (Synchronised Multimedia Integration Language) is a recent open standard for the Internet and allows objects to be called into the browser software at given points. This is a simple form of sequencing and requires the browser to be recent enough to support the technology. SMIL is unable to offer any flexibility over pre-fetching, multi-layering and more importantly, intelligent playback of an arrangement in line with fluctuations in network throughput. SMIL has been devised with a broadband concept in mind as with today's narrowband networks, the performance of an SMIL communication is compromised by the time it takes to pull in the desired file at the given point.

MIDI techniques are also utilised for generating audio. MIDI files, by their nature are smaller than files which attempt store an actual sound wave pattern in digitised format. This means they can be readily transferred over a network faster than other types of audio files. However, MIDI files do not reproduce pre-recorded audio sounds. Instead, a set of instructions following a standard known as GM MIDI is executed by the computer through a sound card



activating notes on particular instruments whose approximate sound characteristics have been stored on the sound cards. The quality of the sound card, or a device attached to a sound card which is capable of accepting a GM MIDI set of instructions, is highly variable and is dependent largely on price. Consequently to obtain realistic sound effects requires expensive

5 pieces of hardware. The reproduction of the audio files is generally of a poor quality, because of the nature of FM synthesis in the "low end" mass-market sound card. Even with a "high end" sound card, the quality of reproduction of audio files is limited to the GM MIDI pre-sets and such pre-sets allow only for basic instrumental sounds which are suitable for limited applications, computer games and the like. MIDI files rely on a sequencing time line but do

10 not require any underlying intelligent management to respond to network variations as the sequence triggers sounds that are already built in to the computer's sound card and not media files that require downloading.

None of the above technologies attempt to create a solution for transferring multimedia

15 or data over a communications network where the media/data is sectioned into segments for transferring over the network and then reconstructed into a composite piece of content that builds 'piece-meal' during a visit to the web site by using any available bandwidth to pre-load the segments ready for replaying according to a time controlled sequence. This sequence dictates the build of the arrangement within a browser where selected segments can be reused

20 either to repeat in the style of the piece or repeat as a loop or play again at a later point to contribute to a play back, that is either continuous or time controlled. The replaying mechanism contains special contingency that enables the reusing of selected files in the event of a subsequent file not being available due to network congestion and extra contingency that allows the properties of selected files to vary according to the states of download and use of

25 the 'file reusing' contingency. This sequencing method allows 'time' to be introduced and exploited within a communication, such as a web site, so as to allow content to be triggered after a measured duration and from a situation where the files are preloaded.

It is therefore apparent that a need exists for a system which is capable of transferring and playing or reproducing media files over a communications network, such as the Internet,

30 which will be compatible with the provision of Web pages and which will provide intelligent



timing and file scheduling to allow content to be built from these component files as they pre-load in a sequential order and where the access time in terms of waiting for the media file to start playing on a user's computer terminal is shortened by the system requiring only one or a selection of files to be present before playing can commence.

5

It is also apparent that a need exists for a system which is capable of transferring and reproducing all types of files over a communications network, such as the Internet, which will make optimum use of the available bandwidth by allowing the download of all said files to be arranged and controlled by a small executable program and a set of sequencing instructions which determines the download order for the entire content of the communication and reproduction order, where the communication is a voice-over, a music composition or other sound track, an audio visual piece containing still images or animated images synchronised to the audio or otherwise and user interaction or otherwise, a piece of streaming media or other multimedia component such as a Flash movie and other compositions such as the components required in a web site. This content is likely to be files of an HTML or similar type, text files, image files, multimedia files and audio files.

It is also apparent that a need exists for a system which is capable of transferring multimedia or data over a communications network where the media/data is sectioned into segments for transferring over the network and then reconstructed into a composite piece of content, such as a musical arrangement or audio/visual piece, that builds 'piece-meal' and during a visit to the web site by using any available bandwidth to pre-load the segments ready for replaying according to a time controlled sequence. This sequence dictates the build of the arrangement within a browser where selected segments can be reused either to repeat in the style of the piece or repeat as a loop or play again at a later point to contribute to a play back, that is either continuous or time controlled. The replaying mechanism contains special contingency that enables the reusing of selected files in the event of a subsequent file not being available due to network congestion and extra contingency that allows the properties of selected files to vary according to the states of download and use of the 'file reusing' contingency. This sequencing method allows 'time' to be introduced and exploited within a



communication, such as a web site, so as to allow content to be triggered after a measured duration and from a situation where files are preloaded.

OBJECT OF THE INVENTION

5 It is an object of this invention to produce an intelligent delivery engine that provides a method of transferring and playing media files over a communications network which will obviate or at least minimise the above disadvantages.

10 It is also an object of this invention to produce an intelligent delivery engine that provides a method of transferring all types of files over a communications network and reproducing said files in such a way as to permit viable use of multimedia over networks where throughput speed is highly variable and a desire is there to be able to time in and intelligently re-use and manage files and adjust their properties accordingly so as to build and deliver a comprehensive piece of content or whole communication, by way of a synchronised
15 delivery which will obviate or at least minimise the disadvantages of current methods of transferring data.

DISCLOSURE OF THE INVENTION

20 In broad terms the invention comprises a method of transferring a media file or other file type over a communications network and reproducing or playing said file, comprising:

- (a) dividing media content into a sequence of segmented files,
- (b) maintaining said segmented files in a provider computer means, and
- (c) transferring in a specific sequence said segmented files to a receiving computer
25 means,

wherein, after each segment has been received by said receiving computer, playing or reproduction of the segment will commence before or during the loading of the next sequential segment,



and further wherein subsequent segments that are received will be played according to sequencing information, wherein the sequencing information dictates which, if any, segment is played and how it is played in the event that the next segment in the sequence is not available for playback or reproduction.

5

Preferably the step (b) further includes maintaining a user loadable program in the provider computer means and step (c) further comprises transferring the program to the receiving computer means.

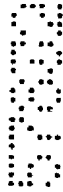
10

Preferably the segment is encoded and is decoded by the user loadable program in the receiving computer means before playback or reproduction.

Preferably the program is a Java applet or an Active X file.

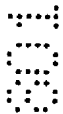
15

Preferably playing of the media file can commence prior to the completion of the reception of the next segment in the sequence.



20

Preferably the sequencing information is parameterised in and transmitted as a text document or other similar type.



Preferably the sequencing information is built into the user loadable program.



25

Preferably the media file is any collection of data such an audio file, a video file, an image file, an animated file, an HTML file, a VRML/3d World file, a text file, a Flash movie, other executable files, instructions for other devices, data streams or a filter which modifies other media.



BRIEF DESCRIPTION OF THE DRAWINGS

One preferred method of transferring and reproducing a media or other file over a communications network will now be described with the aid of the accompanying
5 drawings, wherein:

Figure 1 is a block diagram of the preferred method of the operation of the system;

Figure 2 is a flow chart showing basic system operation;

10

Figure 3 depicts one form of pseudo code for the applet of the preferred embodiment.

Figure 4 is a chart illustrating diagrammatically one stage in the replaying of a
15 segment of a media file.

Figure 5 is a chart illustrating diagrammatically a still further stage in the replaying of a segment of a media file.

20

Figure 6 is a chart illustrating diagrammatically a still further stage in the replaying of a segment of a media file.



clm 10993855v1

Figure 7 is a chart illustrating diagrammatically another stage in the replaying of a segment of a media file.

5 Figure 8 is a chart illustrating diagrammatically another stage in the replaying of a segment media file.

Figure 9 is a chart illustrating diagrammatically another stage in the replaying of a segment of a media file.

10

Where figures 4 – 9 combine to illustrate the intelligent nature of the technology.

BEST MODE OF PERFORMING THE INVENTION

15 The system 2 as shown in general block diagram in Figure 1 comprises a provider 3 including a provider modem 4, a server 5 and provider memory 6 containing a web page 7, a small executable program [SEP] 8 and one or more media files 9 sectioned into segments 10, whose lengths and sizes are determined by the designer or dictated by the nature of the original media where music would normally be sectioned into logical phrases for example.

20 These segments 10 are created as individual files that make sense when played individually or in conjunction with another file as part of a whole piece or multi-layered arrangement. As with a short musical phrase, spoken voice might be separated into single words, sentences or phrases and animations separated into background images and foreground images.

25

The segments 10 can be reusable at any time along the sequence. An example of this might be a musical bar's worth of drums playing a beat or the background image to an animation. The deployment of these reusable segments 10 happens within a user's computer for the purposes of saving resources. When applied to the Internet, these resources might be



bandwidth and download times, where the SEP makes optimum use of bandwidth at any time until all the segmented files are transferred.

The deployment of these reusable segments 10 also happens within a user's computer for the purposes of setting up a timing rate for controlling the playback of the segmented files in the correct manner but by default, this timing can be utilised to control announcements and other timed in events and or content that introduces the true notion of 'time based viewing' to the likes of an Internet Web site or other communication.

10 The provider 3 may be connected to a plurality of users 13 by a communications network 12 such as the Internet. The user system 13 includes a user modem 14 or other network connection device, a user computer 15 and a user memory 16 containing an Internet browser 17. The browser 17 includes an interpreter which interprets and executes the SEP 8 and its sequencing instructions.

15 As illustrated in Figures 1 and 2, a provider computer 3 maintains in the provider memory 6, a web page file 7, an SEP 8 and one or more media files 9, or other files, in the form of encoded files or other file types 10 which may be audio, video, graphical, html, and other known types of files which contributes to the content of the communication over a communications network. The encoded files or other file types are obtained by the SEP 8 from the server 5 and represent the media file 9, complete content, music piece or other arrangement, as a number of sections, each of which is encoded or compressed into an encoded file.

25 The user system 13 can download the web page 7 and the SEP 8 and set of sequencing instructions of the provider 3 by using the Internet 12 and user and provider modem or other network connection device interfaces 14 and 4 respectively. The web page 7 provides the user 13 with the option of downloading and reproducing one or more media files 9. Upon selecting a file 9, the SEP 8 now resident in the user memory (shown in phantom outline) is executed by 30 the user's browser software 17. The SEP 8 is preferably written in Java or Active X for the



initial purpose as an Internet application, although any language interpretable by the browser 17 may be employed as long as it supports media files. Alternatively, the user 13 may use resident executable application software to download and reproduce the encoded files 10.

5 Alternatively the initial web page 7 and the SEP8 of the provider 3 can be downloaded and once downloaded, the SEP 8, executed by the browser software 17 can control the further download of all the file types, media or otherwise, that form the whole of the content of the communications (web pages and their content).

10 The SEP 8 starts downloading the first of the sequential encoded files or other file types 10 [1.1], and waits until this is fully loaded into the memory 16 of the user computer. The SEP 8 then decodes or decompresses the encoded file 10 [1.1] into a decoded file 19 [1.1p] and commences playing or reproducing the file if instructed to do so by the set of sequencing instructions. At the earliest point, normally in conjunction with playing the loaded
 15 files, the SEP 8 then starts loading the next sequential encoded file or other file type 10 [1.2] [1.3], and at the completion of loading each encoded file or other file type 10, the SEP 8 decodes it into a decoded file 19 and can commence playing or reproducing it at such point that the sequence information dictates. In practice the decoded files 19 [1.1p] [1.2p] ... will be added to a queue which will enable each decoded file 19 to be played in a first in, first out
 20 (FIFO) sequence such that one file, for instance [1.1p] runs into the next [1.2p] if required to do so by the sequencing information. The media file or other file type 9 will therefore appear to be played continuously without pauses between decoded files. This system therefore requires a user 13 to supply only a basic sound card 21 (for purposes of audio) and modem 14, or other network connection device, to play a high quality media (audio) file 9 or other file
 25 types. Alternatively the files can be arranged to play or be reproduced according to a sequence where timed spacing is employed. In the event of this, the next file 19 in the queue will still be downloaded at the earliest opportunity and will remain in the memory 16 of the user computer until such time as it is required by the sequencing information. The sequencing information might be parameterised in a text document or built into the executable program.



The timing of the loading of the encoded files or other types 10 and playing of the decoded files 19 can be seen at 20 in Figure 1 where the first decoded file [1.1p] does not start playing until after it's corresponding encoded file 10 [1.1] has been fully loaded and decoded. Loopback points are defined within the sequencing information, indicating suitable phrases to repeat in the event of being unable to progress further in the sequence (typically, although not necessarily because a required media file or other file type 10 is not yet completely downloaded 10 or decoded 19). The presence and availability of loopback points give the impression of continuous output for purposes of achieving a continuous flow. In addition to loopback points, other contingency events and instructions are used to add greater control to the playback of the files and in particular to sustain the reproduction. 'Play once' commands are applied to files that should not repeat in the event of a loopback happening in an area where the said file resides. 'Play once' is used for files such as voice files, where repeating the voice during a loopback would seem strange. 'Needed by' commands are applied to files so as to group more than one file together to retain exact phrasing. 'Needed by' can also be used to stimulate a loopback point in a position that is better suited to the arrangement than the default trigger point, governed by the 'play' position of the missing file.

The user 13 has only to wait for the loading and decoding of the first encoded file or other file type 10 [1.1] and does not have to wait for the loading of the complete media, complete content, music piece or other arrangement, or other file types before the media file 9 or other file type commences playing or is reproduced. The partitioning of one complete media file removes the need for the complete media file to be loaded prior to playing.

Preferably the Java SEP 8 comprises two elements or "threads" which run simultaneously during the life span of the Java SEP (as shown in Figure 3). The first element is the kernel or loader which starts the second element and loads the encoded files 10 from the provider 3. The second element is a player/sequencer which sits in a loop continuously monitoring the state of the encoded and decoded files 10 and 19. During the loop, if the player/sequencer detects that a decoded file is available for playing (i.e. and encoded file has been loaded and decoded) it will play the file (at the start of the loop to maintain



synchronisation) as long as the file obeys a set of rules defined for it. This set of rules defines the sequencing of the files. This element also maintains a counter which represent the position in the media file 9. This counter, combined with the check for file playability and the logic of the sequencing rules, allows the SEP to intelligently sequence the decoded files providing an effective sequencing unit.

Referring to Figures 2 and 3 when the SEP 8 is executed, the kernel initially loads the sequence information about the media file(s) 9 to be downloaded, then starts loading the first encoded file or other file type 10. The kernel then initialises any media systems required such as sound cards or video playback systems and creates and/or starts the player and loader threads running such that both elements run simultaneously.

The sequencing information is timed by way of beats (timed measures), each allowing for a set of events which happen within the beat. Beats happen at regular distinct intervals defined within the sequencing information. A beat can be given a different value at a specific point within the sequence as an event. This allows a combination of media files of differing lengths and rhythms to be used within the same arrangement. Events are actions that can be performed by the player. Some potential actions are:

- start or stop playing a media file
- alter the contents of a media file (e.g.: blurring an image, applying reverb to a sound)
- setting properties of a media file (e.g.: setting the level of fog in a 3D world).
- stopping playback
- altering the next beat to be played (e.g.: jumping, repeating sections)
- act on input from outside the player. For instance, input from the user, or from a coexisting piece of software or from a peripheral device attached to the computer, could result in the player performing one or more actions (events). The input need not arrive at the same point as the event is processed by the player, but could be received earlier and stored until needed.



- synchronous control – in addition to events which act on external input, the player itself can respond directly and immediately to specific commands. These might include the ability to suspend playback (pause), or to disable and re-enable specific types of events.
- 5 • synchronising of sound files and image files.
- synchronising of sound files and HTML pages.
- synchronising the timed delivery of any piece of content for an Internet web site

After an encoded file or other file type 10 has been loaded by the loader thread, the
 10 player decodes the encoded file 10 into a corresponding decoded or playable file 19. The loader, at the earliest opportunity, starts loading the next encoded file 10 in the sequence.

Meanwhile the player loads the first decoded file 19 [1.1p] required for the start of the
 sequence. The player then sits in a loop receiving instruction from the sequencing information
 15 loaded by the kernel. If the player is able to perform the events contained in each beat as instructed by the sequencing information, it does so while scanning through the next beat to ensure the events contained in that next beat are able to be performed. If the events in the next beat can not be performed because the next encoded file or other file type 10 has not been
 20 fully downloaded or decoded, the player sets the next beat to be performed to be the last encountered loopback point or other point as dictated by the sequencing instructions. After finishing one beat, the player waits for the next beat to load and repeats this cycle unit all available beats in the sequencing information have been performed on the decoded files 19.

As a result of the present invention, many of the deficiencies previously inherent in the
 25 transmission of media files over transmission networks have been minimised including the problems of transmitting and reproducing all file types in a predetermined order over transmission networks where the network throughput speed is unstable. In particular some of the advantages are:

- 30 a. Special decoding software does not have to be installed by the user.



- b. Any available bandwidth can be used for transmission of a media files and components of an arrangement, such as a piece of music, unlike live stream transmission.
- c. The user does not have to wait for the whole media file or content, such as a piece of music or complete spoken paragraph, to be downloaded some before playing commences.
- d. Actual pre-recorded sound waves are reproduced and not computer generated sounds.
- e. Standard readily available hardware can be utilised by the receiver.
- f. Any combination of files can be sequenced and thus the arrangement of the download of the files can be pre-designed to optimise the resources of the bandwidth.
- g. Multi-media content and other communications can build 'on the fly' during the visit so as to remove the need to send down complete or near complete arrangements and to allow much larger compositions to be deployed.
- h. Time factors can be introduced and deployed within a communication, such as a web site, to allow the content to be triggered at points designated by the designer, along an independent time line.
- i. Compositions can be placed inside an existing web site in such as way as to tie them to specific pages

EXAMPLES

The download and playback of the segments of data is monitored by the SEP 8 using a set of sequencing instructions such that certain contingency events are triggered according to the network performance and how it affects the playback of said sequence. These functions are deliberate and implemented within the sequencing instructions by the designer and are to enable:

Figure 4 illustrates diagrammatically the replaying of an individual file or collection of files once or as a phrase in the event of the next required file segment not being ready for replaying. An example of the might be when applied to the Internet, that a segment file "a" is arranged to play for 8 measures. However, after 4 measures, a second file "b" is arranged to play in conjunction with file "a". In this example, we can assume that file "b" has not



completely downloaded or arrived at the desired place from where it can be utilised. In this event, the contingency would be that file "a" was replayed an extra "x" number of times until file "b" had arrived where "x" is one pass or a number of passes of a section chosen by the designer. Once file "b" has arrived, the sequence would continue.

5

Figure 5 illustrates the replaying of an individual file or collection of files once or as a phrase in the event of the next required file segment not being ready for replaying. An example of this might be when applied to the Internet, that a segment file "c" is arranged to play once. However, once it has completed it's playing, a second file "d" is arranged to play also once, at a given point sometime beyond file "c" and at a time designated by the designer. These two files might be individual spoken sentences and would not be suitable for repeating. In the event, the contingency would be that once file "c" had completed playing, the sequence would replay a silent measure or number of silent measures between c & d until file "d" has arrived and can be played.

15

Figure 6 illustrates the replaying of an individual file or collection of files once or as a phrase in the event of the next required file segment not being ready for replaying but where some of the files included in the repeatable selection are not suitable for repeating. An example of this might be when applied to the Internet, that a segment file "e" is arranged to play for 8 measures. However, after 4 measures, a second file "f" is arranged to play in conjunction with file "e". In this example, we can assume that file "f" is of a type that is not suitable for replaying, such as a spoken sentence which, if repeated would sound unusual. A third file "g" is arranged to join file "e" at a time after the playing of file "f", either individually or in conjunction with file "e". In this example, we can assume that file "g" has not completely downloaded or arrived at the desired place from where it can be utilised. In this event, the contingency would be that file "e" was replayed an extra "x" number of times until file "g" had arrived, where "x" is a number chosen by the designer and where file "f" is marked/noted in the arrangement, as a file type that plays once only and does not replay, if the part of the sequence that it resides in is re-used for whatever reason. Once file "g" has arrived, the sequence would continue.

30



Figure 7 illustrates the replaying of an individual file or collection of files once or as a phrase in the event of the next required file segment not being ready for replaying, where this contingency is triggered at a time prior to the playing time in the arrangement of the still, as yet loaded file. An example of this might be when applied to the Internet, a second file "I" is arranged to play in conjunction with file "h". In this example, we can assume that file "I" has not completely downloaded or arrived at the desired place from where it can be utilised. In this event, the contingency would be that file "h" was instructed to replay an extra "x" number of times, by a special event placed within the arrangement by the designer, at a point where the replaying of file "h" would start at a time deemed more useful than from after measures, such as after measures for instance, which would be musically more appealing. Once file "I" has arrived, the sequence would continue.

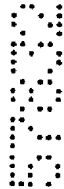


Figure 8 illustrates the replaying of an individual file or collection of files once or as a phrase, in the event of the next required file segment not being ready for replaying, where this contingency is triggered at a time prior to the playing time in the arrangement of the still, as yet loaded file. An example of this might be when applied to the Internet, that a segment file "j" is arranged to play for a single measure. However, after a further 2 measures, a second file "k" is also arranged to play. In this example, we can assume that file "k" has not completely downloaded or arrived at the desired place from where it can be utilised. In this event, the contingency would be that file "j" was instructed not to play, by a special event placed within the arrangement by the designer until file "k" had too arrived. This special event would pre-fetch the file "k" before either starting the sequence, or in the event of this situation occurring at a later point in the arrangement, continuing with the sequence, thus ensuring that the time interval between the two files in question was assured and controlled by the sequence clock, as set by the designer. Once file "k" has arrived, the sequence would continue by playing both files "j" and "k".



Figure 9 illustrates the playing and replaying of an individual file or collection of files as seen in all the examples above but where the arrangement pauses or concludes for a pre-



determined amount of time, at a point selected by the designer, but then continues by either reusing files already downloaded or playing new files or a combination of both. An example of this might be when applied to the Internet, that a portion of an arrangement or sequence had played to a point from where the designer had decided to pause for an amount of time. The time here would be controlled by inserting an appropriate number of silent measures of known time intervals or by inserting an adjusted time measure of known time value into the arrangement. The next batch of segmented data files ahead of the time that the designer had arranged them to replay so as to optimise on the performance of the arrangement. This event allows the designer to introduce the notion of time to say an Internet page an audio file audibly states that some event will occur after an amount of time.

The synchronising of data of different types such as image files, HTML documents, animated images, other executable files (other applets), instructions for other devices, data streams and audio file, where any one of the above contingency events can occur so as to permit the delivery of the arrangement/sequence over a network such as the Internet, when the network throughput/bandwidth is subject to fluctuations in performance. An example of synchronising of different file types might be, when applied to Internet advertising: an audio sequence is pulled into the HTML of a web page. The said web page is positioned within the site so as to persist throughout the visit to the site. Contained within the sequence is a collection of image files as well as the audio files that make up the background sound. These images are the image files that are later to be used by a subsequent HTML page. This second HTML page is also placed within the sequence and triggering of this page is controlled by the sequencer, such that it will not be played until the image files and any other files it requires are already downloaded, by the sequencer. At the correct moment, when the necessary files are downloaded, the sequencer will move along to the point where it triggers the second HTML page and targets this page to a pre-chosen area on the screen. The effect of this is to automatically synchronise the delivery of the HTML page with the playing and showing of the audio and image files contained within it. As this method is controlled by the original sequence, the HTML page can then be switched back to the previous page or a third page can be synchronised in, in the same way. The intention here is to use the engine to create a new



method for advertising on the Internet which is not only audio/visual but can be timed in and out again just as a television commercial break might be.

5 The sequencing instruction also may carry instructions that cause variable repeats to occur, or timed repeats and jumps, where the data played in the repeated section is already downloaded or is a combination of already downloaded data and new data. Variable repeats might be in the style of DS, Fine, Da Capo, Coda etc as seen in written music. An alternative example of variable repeats might be when using the engine to deliver animations, where for the first play through a different background image is used to that on the repeated section but
10 whose foreground images are the same and re-used.

The sequence timing may also be of a varied and changing value throughout the arrangement. The purpose of this is to: control the speed at which the files are replayed, allow files of different musical styles to be used within the same arrangement, create play points part
15 the way through an existing file by dividing the sequence time by a chosen amount, allow the timing in of content at a given point in the future of the communication and other such things.

The foregoing describes a preferred form of the invention. Having read the description it will be apparent to those skilled in the art that alterations and modifications can be made without departing from the basic concept of the invention. All such alterations and
20 modifications are intended to be incorporated within the scope hereof.

The word 'comprising' or forms of the word 'comprising' as used in this description and in the claims do not limit the invention claimed to exclude any variants or additions.



CLAIMS

1. A method of transferring a media file over a communications network and reproducing or playing said file comprising:

5

- (a) dividing said media file into a sequence of segmented files,
- (b) maintaining said Segmented Files in a provider computer means, and
- (c) transferring in a specific sequence said segmented files to a receiving computer means,

10

wherein, after each segment has been received by said receiving computer, playing or reproduction of the segment will commence before or during the loading of the next sequential segment,

15

and further wherein subsequent segments that are received will be played according to sequencing information, wherein the sequencing information dictates which, if any, segment is played and how it is played in the event that the next segment in the sequence is not available for playback or reproduction,

20

2. The method of claim 1, wherein the step (b) further includes maintaining a user loadable program in the provider computer means and step (c) further comprises transferring the program to the receiving computer means.

25

3. The method of claim 1, wherein the segment is encoded and is decoded by the user loadable program in the receiving computer means before playback or reproduction.

4. The method of claim 1, wherein the program is a Java applet or an Active X file.

30

5. The method of claim 1, wherein playing of the media file can commence prior to the completion of the reception of the next segment in the sequence.



6. The method of claim 1, wherein the sequencing information is parameterised in and transmitted as a text document or other similar file type.
- 5 7. the method of claim 6, wherein the sequencing information is built into the user loadable program.
8. The method of claim 1, wherein the media file is any collection of data such as an audio file, a video file, an image file, and animated image file, an HTML file, a
10 VRML/3D World file, a text file, a Flash movie, other executable files, instructions for other devices, data streams or a filter which modifies other media.

Tangozebra Limited

August 22, 2001

B
P
ES
E

N



clrm 10993855v1

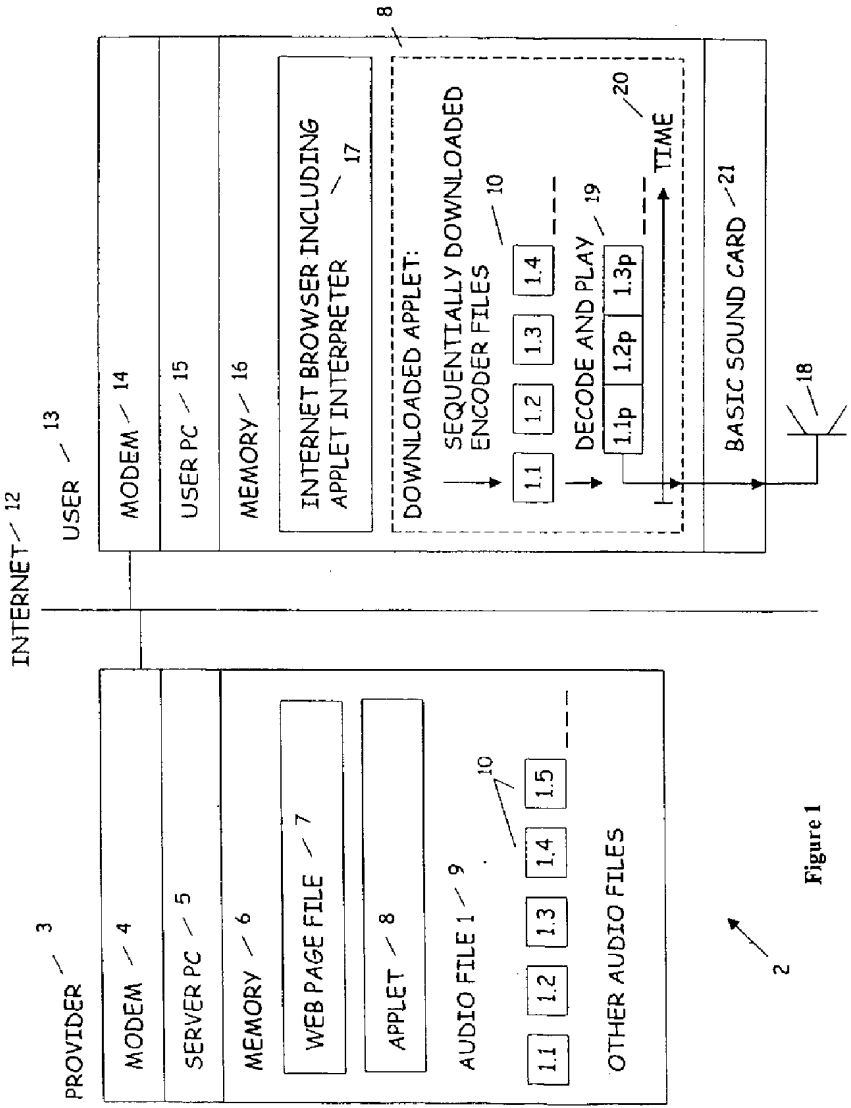


Figure 1

2/3

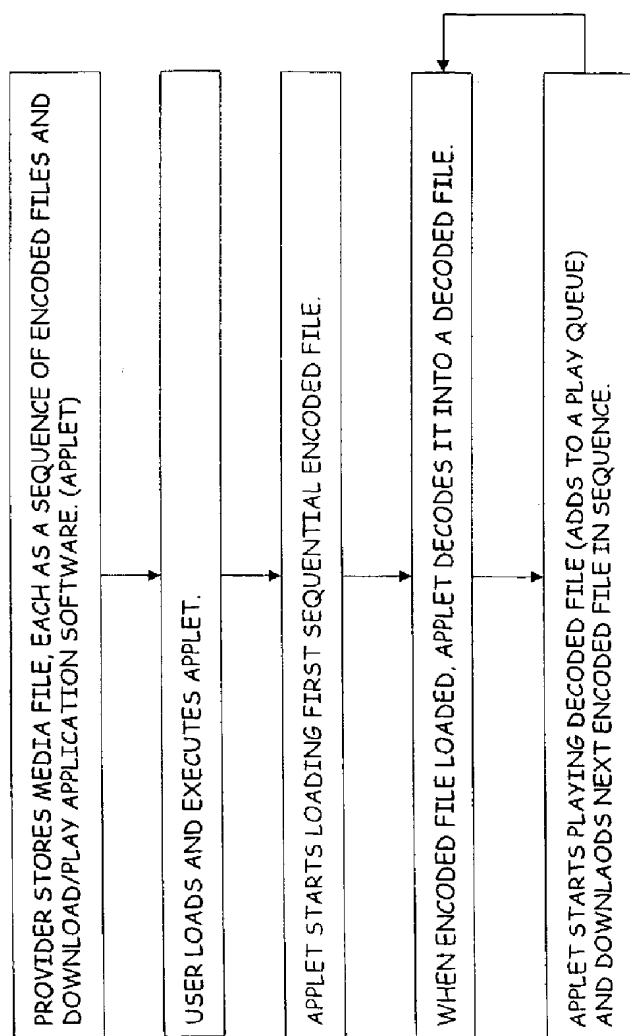


Figure 2

3/3

FIGURE 3

Pseudo code for kernel/loader

```

{ Init }
  Load sequence information, including list of media files
  Create and start player thread
{ End init }

{ Main code }
  FOR EACH media file DO
    if file not yet started loading
      start file loading
    if file not yet finished loading
      wait until file has finished loading
  end FOR EACH
{ End main code }

```

Pseudo code for player/sequencer

```

{ Init }
  Start loading media files required for the start of the sequence
  Wait until the above are all loaded
  Initialise and media systems required (eg: sound or video playback)
  Start the loader thread (runs the Main code section above)
  Start the player thread (runs the Main code section below)
{ End init }

{ Main code }
  FOR EACH beat in the sequencing information DO
    FOR EACH event in the beat DO
      perform the event (this may involve stopping playback, or altering
        which beat is to be performed next)
    end FOR EACH

    scan through the next beat, ensuring that all events in the beat are
    ready to be performed

    if this is not the case
      set the next beat to be performed to be the last encountered
      loopback point

    wait until it is time to start playing the next beat in the sequence
  end FOR EACH
{ End main code }

```